



(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : **94308394.9**

(51) Int. Cl.⁶ : **G06F 3/033**

(22) Date of filing : **14.11.94**

(30) Priority : **16.11.93 JP 286387/93**

(43) Date of publication of application :
17.05.95 Bulletin 95/20

(84) Designated Contracting States :
DE FR GB

(71) Applicant : **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504 (US)

(72) Inventor : **Hidaka, Kazuyoshi**
2-177, Fujidana-cho,
Nishi-ku
Yokohama, Kanagawa-ken (JP)

(74) Representative : **Williams, Julian David**
IBM United Kingdom Limited,
Intellectual Property Department,
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) **Cursor with multiple pointing spots.**

(57) A mouse cursor is used to selectively subject an object displayed on a screen (3) to one of a plurality of operations with a minimum number of steps.

Positioning the first pointing spot of a multiple point cursor with respect to an object and clicking a mouse button (9a, 9b) designates, for example, the first process, namely, the transfer operation of the object. In response to the clicking of the mouse cursor after the transfer of the cursor, the transfer operation is performed such that the object in the original position is deleted and a designated object appears in the current cursor position. The second position of the multiple cursor is then positioned by clicking the mouse cursor with respect to the object so as to designate the copying operation of the object. In response to the clicking of the button after transferring the cursor, the copying operation is performed such that the designated object appears at the current position.

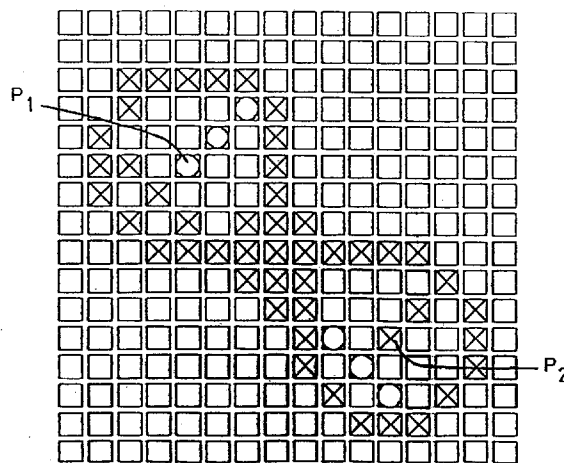


FIG 10

The present invention generally relates to data processing systems having a display and a pointing device such as a mouse or a track ball, and more particularly to a data processing system in which a pointing device indicates an object displayed on a display and performs a predetermined operation on the object.

Conventionally, with a computer system having a graphic user interface (GUI), it has been common to perform a predetermined operation such as execution of a program associated with the icon, and expansion and transfer of a window by positioning a mouse cursor on an object displayed on a display such as an icon, a graph, a window frame, or a character to click, double click, or drag a mouse button.

Along with the complication of GUI application programs, operations that should be performed on an object on a display have become diversified. To cope with the diversification of such operation, an attempt is made in the prior art to improve the user interface and to improve the efficiency of the operation.

PUPA 59-163666 discloses a graphic generator using multiple cursors which forms a cursor area in an area surrounding part of a graphic pattern, transfers in a voluntary direction a graphic surrounded by the cursor area, and further adds or contracts the graphic along with the transfer of the graphic.

PUPA 62-11927 discloses a calculator using an icon for display and selection of functions in which the functions possessed by each icon, its position on the screen, the icon shape, and menu data displayed when an icon is displayed are stored and controlled to display the content of menu data when an icon for controlling the cursor display is selected.

PUPA 62-259129 discloses the provision of an indication means for indicating voluntary two pointing spots with a relative distance therebetween on a tablet and cursor control means for transferring a graphic cursor displayed on a display following the transfer of the graphic cursor between the two pointing spots as well as change of the size of graphic cursor in response to the pressing down of a predetermined button.

PUPA 2-166506 discloses an information input apparatus that displays a cursor on a position on a display screen of a display in response to the operation of a pointing device, the cursor serving as a shape for representing the function of operation means.

PUPA 4-34663 discloses the provision of a function for representing the scope of an object to be selected with a cursor for selecting and editing an object to be edited on a display, as well as allowing the variable setting of the shape and size of a cursor for representing the scope of an object to be selected.

The above prior art discloses change of the shape and size of a mouse cursor in accordance with a different operation, the cursor serves only to select

one object. Consequently, when an attempt is made to perform one of a plurality of possible operations, for example, the following processing steps will be followed.

(a) A specific object is selected by positioning a mouse cursor on the specific object and clicking a mouse button. At this time, the specific object changes color, inverts its display, and becomes shaded to indicate that it has been selected. However, the operation that is to be performed has not been established, therefore, the operation is not performed on the object at this stage.

(b) Then the menu on the upper part of the window is clicked to display a pull-down menu.

(c) A desired operation menu is clicked from among a plurality of operation menus displayed in the displayed pull-down menu. As a consequence, the desired operation is performed on the selected object.

Such operation, however, requires many operation steps using a mouse cursor. The operation is thus inefficient. There is known an attempt of performing a specific operation with a combination of a control key and the right button of the mouse as a short cut key to cope with the above operation. Users are obliged to memorize a specific key stroke. Such an attempt is also lacking in flexibility on a user interface.

Accordingly, viewed from one aspect the present invention provides a data processing system having a processor, a display coupled to the processor and a pointing device coupled to the processor such that movement of the pointing device results in corresponding movement of a cursor on the screen of the display, the data processing system comprising: means for displaying a cursor having first and second pointing spots on the screen of the display; means for instructing the processor to perform a first operation represented by an object displayed on the screen when the first pointing spot of the cursor is positioned on that object; and means for instructing the processor to perform a second operation represented by an object displayed on the screen when the second pointing spot of the cursor is positioned on that object.

The present invention provides a method for processing data and an apparatus thereof in which a mouse cursor is used on an object displayed on a display screen to enable one of a plurality of operation to be selectively performed with a minimum operation step.

This is attained by using multiple pointing spots cursor on an object in accordance with the present invention. In other words, the transfer operation of the object, for example, is designated by positioning the first pointing spot of multiple point cursors and clicking the mouse button. In response to the clicking of the mouse button following the transfer of the cursor, the transfer operation is performed in which an object at the original position is erased in response to the

clicking of the mouse button following the transfer of a cursor, and a designated object appears on the current position of the cursor. Then a copying operation, for example, is designated to an object by positioning a second pointing spot of a multiple point cursor and clicking the mouse button. In response to the clicking of the mouse button following the transfer of the cursor, the copying operation is performed in which a designated object appears at the current position of a cursor.

Furthermore, the system can dynamically change the shape of a multiple point cursor, the number of pointing spots depending on different applications and necessary operation.

In order that the present invention may be fully understood preferred embodiments thereof will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a view showing a hardware construction for using the present invention;

Figure 2 is a view a block diagram showing a system construction related with the present invention;

Figure 3 is a view showing an example of an object displayed on a screen;

Figure 4 is a view showing an object table;

Figure 5 is a view showing an example of the multiple point cursor;

Figure 6 is a view showing an example of the multiple point cursor;

Figure 7 is a view showing an example of the multiple point cursor;

Figure 8 is a view showing an example of the multiple point cursor;

Figure 9 is a view showing an example of the multiple point cursor;

Figure 10 is a view showing a bit map and pointing spot in the multiple point cursor;

Figure 11 is a view showing an example of a mouse information control table;

Figure 12 is a view showing a flowchart of the control operation of the multiple point cursor;

Figure 13 is a view showing a flowchart of the control operation of the multiple point cursor;

Figure 14 is a view showing flowchart of an example of the application program using an operation of Figures 12 and 13;

Figure 15 is a flowchart for the operation of the vertical bisector using the multiple point cursor;

Figure 16 is a view showing the state of vertical bisector operation using the multiple point cursor;

Figure 17 is a view showing a flowchart for an object copying and transfer operation using the multiple point cursor;

Figure 18 is a view showing a flowchart for an object copying and transfer operation using the multiple point cursor;

Figure 19 is a view the state of bisector operation

using the multiple point cursor; and

Figure 20 is a view showing in detail the multiple point cursor used in the operation of Figure 19.

Figure 1 is a view showing an external appearance of a hardware construction which runs a system in accordance with the present invention. The system shown in Figure 1 comprises a system device 1 incorporating a CPU, a RAM, a hard disk drive, a floppy disk drive, a graphic display device 3 coupled to the system device 1 having an APA (all point addressable) video buffer, a keyboard 5 coupled to the system device 1 via a cable 7 and a mouse 9 coupled to the system device 1 via a cable 11. The mouse 9 is provided with a left button 9a and a right button 9b which are pressure-operated. The left button 9a, which depends upon an application program using the mouse 9, is usually used in the selection of an object such as clicking, double clicking, and execution whereas the right button 9b is used for erasure.

The system device 1 shown in Figure 1 operates under a voluntary operating system with functions for setting mouse cursor position acquisition and mouse cursor configuration such as OS/2 (registered trademark of IBM), PC-DOS (registered trademark of IBM), MS-DOS (registered trademark of Microsoft Company), WINDOWS (registered trademark of Microsoft Company), X-WINDOW System (registered trademark of MIT), a window system that is actualized under AIX (registered trademark of IBM) and UNIX (registered trademark of UNIX System Laboratories), or System 7 of Apple Computer Company.

Figure 2 is a block diagram showing a construction for realizing the system of the present invention. Referring to Figure 2, there is provided in the system device 1 a bus 120 for exchanging data. A known processor (CPU) 103 having a function of controlling calculation, memory, input, and output is connected to the bus 120. The mouse 9, also shown in Figure 1, is connected to a bus 120 via a mouse controller 105 to serve for detecting the revolution of a roller (not shown in the drawings) provided on the bottom of the mouse 9 and the pressing of buttons 9a and 9b (Figure 1) to transmit a detection signal to the CPU 103 via the bus 120.

The display 3, also shown in Figure 1, is connected to the bus 120 via a display controller 107. Since the construction is known, the detail thereof is not shown. The display controller 107 is provided with an APA video buffer, a color palette register, and a DAC. The CPU 103 decodes a command described in a program to read and write a video buffer value and set a value to the color palette register, thereby controlling graphic description on the display 3.

The memory 113 is, for example, a random access memory (RAM), which is a region where the above operating system is loaded when the power source of the system device 1 is turned on. It is also a region where the control of an operator or the con-

trol of the CPU 103 loads an application program from a hard disk drive (not shown in the drawings), floppy disk drive (not shown in the drawings), or CD-ROM drive (not shown in the drawings).

In the memory 113 an object control table 115 and a mouse information control table 117 are prepared and maintained by the application program in which the mouse is used.

The multiple point changing apparatus 109 and the multiple point image preparing/changing apparatus 111 are depicted as a block different from the memory 113. Each of these apparatus is preferably an application program loaded on the memory 113.

The outline of the operation of the system in Figure 2 is shown below. In other words, when the power source of the system is turned on, the operating system is read from a hard disk to the memory 113. With such an operation, the CPU 103 follows the control of the operating system, thereby controlling the display controller 107 for displaying on the display 3, for example, the initial screen of the multi-channel window.

Subsequently the operator applies a mouse cursor on a predetermined icon on the screen and actuates an application program using a multiple point mouse function of the present invention by double clicking a right button 9a of the mouse 9. The program then opens one new window to display a predetermined menu and to prepare an object control table 115 and a mouse information control table 117 on the memory 113. The CPU 103 follows the content of the mouse information control table 117 to display the multiple point cursor on the window.

When the operator transfers the mouse 9 on the table, a ball (not shown in the drawings) provided on the bottom of the mouse 9 rotates. A mouse controller 105 detects this rotation of the ball, and the detection signal is input to the CPU 103 through the bus 120. The CPU 103 then changes the display position of the multiple point cursor on the display 3 via the display controller 107 (in actuality, the image of the mouse cursor is erased once and depicted again).

Then, when the operator presses down (clicks) the right button 9a of the mouse 9, the mouse controller 105 detects the press-down operation and transmits to the CPU 103 a signal notifying the detection thereof. The CPU 103 then refers to the object control table 115 in the memory 113 and identifies an object at which the mouse cursor is positioned. At the same time, the CPU 103 refers to the mouse information control table 117 and identifies whether any of the multiple pointing spots out of the currently used multiple point cursor is positioned on the object, and performs a predetermined operation in response to the pointing spot identified.

The multiple point changing apparatus 109 is provided with a function for changing the number and arrangement of multiple pointing spots in the multiple point mouse cursor in response to a command from

the CPU 103.

The multiple point changing apparatus 111 has the function of a normal bit map graphic editor, but it also has an additional function for designating multiple pointing spots.

Next, the specific structure and logic for realizing a multiple point cursor will be explained.

Object Control Table

In the present invention, there are graphic display elements such as lines, squares, circles, letters, and icons. For example, when an application 1 displayed in the window 3 is considered as a reference on a screen of the display 3 shown in Figure 3, a triangle G1, a circle G2, and lines G3 and G4 are individual objects. When the operating system serves as a reference, windows 3a and 3b, and icons 3d, 3e, 3f, and 3g are individual objects.

Referring to Figure 4, an example of an object control table 115 is shown. The entry of the object control table 115 consists of an object ID and substantial data associated with the object ID. The kinds of objects are diverse. Since it is difficult to prepare a fixed entry field, the object control table 115 stores nothing but a pointer for object construction. Individual object construction data stores kinds of objects (circles, lines, squares, and letters), one or more sets of X, Y coordinate data, and attribute data such as color or the like. Their contents are dynamically changed by the operating system and the application program. In addition, the object control table 115 itself is changed by the operating system and the application program so as to dynamically add or delete the object entry. In an example shown in Figure 3, the object control table for windows 3a and 3b, and icons 3d, 3e, 3f, and 3g is globally controlled by the operating system, whereas the object control table for objects G1 through G4 is locally controlled by an application 1. By the way, referring to Figure 3, the object ID is designated with a serial number beginning from one. This is merely an example and it may be anything as long as it identifies an object in some sense. It is not restricted to such a value.

Multiple Pointing spot Mouse Cursor

An example of a multiple point mouse cursor is shown in Figures 5 through 9. At the outset, Figure 5 shows an example of a double point mouse cursor. As shown in Figure 5, the multiple point mouse cursor typically consists of one reference point (pointing spot 1 in the case shown in Figure 5) which itself is a pointing spot, and the offset X component and Y component of different pointing spots to the reference point. Figure 5 shows offset X component and Y component as Δx_2 , and Δy_2 , respectively.

Figure 6 also shows an example of a mouse cur-

sor having a larger number of pointing spots. In this case, pointing spot 1 shown in Figure 6 serves as a reference point, and the offset to the i-th pointing spot is given as Δx_i and Δy_i ($i=1$ through n , n represents the number of pointing spots).

Here, the acquisition function of the coordinate position is incorporated in a voluntary operating system supporting a mouse cursor function. For example, MS-DOS and PC-DOS stores a value such as 03H in an AX register. When the software of INT33H is interrupted, the X coordinate is stored in a CX register and the Y coordinate in a DX register. In addition, an API function called MouGetPtrPos is used in OS/2. The X coordinate is acquired in an element named row of the structure of _PTRCOL and the Y coordinate is acquired in an element called col. Addition of offset to the i-th pointing spot to the coordinate values thus acquired determines the current coordinate values X_i and Y_i of the i-th pointing spot. In addition, in these examples, the reference point itself is given as the pointing spot, but the reference point itself may not be the pointing spot. When the reference point is given as the pointing spot, the formula of $\Delta x_1=0$, $\Delta y_1=0$ is established. When the reference point is not given as the pointing spot, the formula of $\Delta x_1 \neq 0$, $\Delta y_1 \neq 0$ is established.

Figures 7 (a), (b), and (c) show various states of a double point mouse cursor. In addition, Figure 8 shows an example of a quadruple point mouse cursor.

Figure 9 shows an example of a double point mouse cursor, the difference between the example and the counterpart shown in Figure 7 being that the figure associated with the operation is represented with a bit map. This double point mouse cursor is used for drawing a line vertically dividing into two a line mentioned later. Consequently, at the position of P1, it is shown in the figure that the horizontal portion of the vertical bisector is designated. At the position of P2, it is shown in the figure that the vertical portion of the vertical bisector is designated.

Figure 10 is a view showing a bit map of the double point mouse cursor in Figure 9. Referring to Figure 9, for example, Symbol x designates a black pixel whereas Symbol o designates a red pixel. Such a bit map is formed by the multiple point image preparing apparatus 111 in Figure 2. In this sense, the multiple point mouse cursor 111 has the function of a normal bit map editor. The multiple point image preparing apparatus 111 has the functions of designating reference position of the multiple point mouse cursor, the number of the multiple pointing spots, and the position of the multiple pointing spots. In the actual operation of the multiple point image preparing apparatus 111, the operator designates the reference points at the outset after the preparing of the cursor bit map. In an embodiment where the reference point itself serves as the pointing spot, the number of the multiple pointing spots is one. Then, in Figure 10, P1 is se-

lected as a reference point or the pointing spot. When the editing is terminated at this stage, a mouse cursor is produced having only one conventional pointing spot. When a different pointing spot (for example P2 shown in Figure 10) is selected by continuing the editing, the offset of the different pointing spot is selected by continuing the editing, and the offset of the X coordinate and the Y coordinate from the reference point P1 up to point P2 is passed to the predetermined buffer region. Then the number of the multiple pointing spots increases to two. When editing is completed at this stage, data such as the number of pointing spots, the reference position of individual pointing spots and cursor bit map are mutually associated and stored on a disk.

Mouse Information Control Table

Figure 11 shows a mouse information control table. The mouse information control table is controlled by the operating system or the application program. With respect to each unit of mouse cursor information, there is held the cursor ID, the number of multiple pointing spots, the coordinate position of the reference points in the bit map, offset values Δx_{ij} , Δy_{ij} of X and Y coordinates from the reference points of the first to q-th multiple pointing spots and the pointer to the cursor bit map data. That is to say, when the number of multiple pointing spots assumes a value r which is smaller than q , the offset value described in the field ranging from $r+1$ to q designates "don't care." In Figure 11, it is represented by [-]. When a pair of offset X and Y values of a series of pointing spots is linked to arrange the NULL pointer in the anchor, there is no need to specifically record the number of pointing spots or preparing the field for the offset of the q set.

Such a mouse information control table can be preferably prepared upon its system startup in the memory 113 (shown in Figure 2) by reading predetermined information from the disk with the operating system. However, the mouse information control table works so that a specific application using a multiple point mouse cursor is constituted in a local memory region of itself at the time of its startup.

In a specific window or application, a double point changing apparatus 109 (shown in Figure 2), which serves as one program loaded in the memory 113 designates which of the cursors is used in accordance with the operation by the operator or a command from the operating system or change the designation thereof. Furthermore, the double point changing apparatus 109, in accordance with the operation by the operator or a command from the operating system, reads new record of a mouse cursor from a disk to add to the mouse information control table or deletes unnecessary record of the mouse cursor from the mouse information control table. Incidentally, refer-

ring to Figure 11, the multiple point mouse cursor ID is represented in a serial number beginning from 1. This is merely an example, and it may individually designate a record for describing data of individual multiple point cursor in one sense. They are not limited to the above value.

Furthermore, the AX=09H in INT33H of MS-DOS has a function for setting the shape of the cursor. It is possible to designate the reference position where the (BX, CX) point in the bit map of the cursor shape data is obtained with a hot spot, or INT33H, AX=03H if BX=cursor hot spot X position, CX=BX=cursor hot spot Y position are designated and an address where the cursor shape data exists is designated to ES:DX in advance. In OS/2, there exists an API called MouSetPtrShape, which has the same function. Consequently, the multiple point changing apparatus 109 called such functions with reference points x, y shown in Figure 11 and sets the multiple point mouse cursor.

Control for Multiple Pointing spot Mouse Cursor

Control operation for the multiple point mouse cursor will be detailed hereinbelow in conjunction with Figures 12 and 13. This control operation routine is called by an application program using the multiple point mouse cursor of the present invention when required. Thus it is possible to realize the routine as a standard API.

At step 1202 shown in Figure 12, information on the currently used multiple point mouse cursor such as the number of multiple pointing spots including positional information on the offset of such pointing spots is acquired from the mouse information control table 117. This operation is actually performed through the multiple point changing apparatus 109. That is because the multiple point changing apparatus 109 incorporates information on which mouse cursor is currently in use.

At step 1204, the bit map of the mouse cursor is displayed at the current cursor position based on information acquired from the mouse information operation table 117. At steps 1206 and 1208, the mouse cursor moves on a screen with the rotation of a mouse ball (not shown in the drawings). INT33H and AX=0BH are used in the mouse cursor movement on MS-DOS. This enables the acquisition in CX and DX from the previous read-out a distance in which the mouse cursor has moved either in the vertical or horizontal directions. Then the position of the mouse cursor can be set by setting in CX and DX desired coordinate positions so as to use INT33H and AX=04H. In OS/2, the position of the mouse cursor can be set with API such as MouGetPtrPos and MouSetPtrPos. At step 1210, the presence of the clicking of the left button (refer to 9a in Figure 1) of the mouse is checked. When the absence of clicking is detected, other application operation is performed. Thus the operation re-

turns from the mouse control function. On MS-DOS, the clicking of the mouse can be detected with INT33H and AX=05H. In OS/2, the clicking of the mouse can be detected with API such as MouGetEventMask. Incidentally, the rotation of the mouse ball and the click of the mouse can be individually detected at steps 1206 and 1208 in Figure 12. API such as MS-DOS, Windows and OS/2 can simultaneously detect the mouse ball rotation and the clicking thereof. They can thus be substituted with a single judgment step at step 1206 and 1208.

Moreover, with respect to Figure 12, except for step 1202, the above control function can be realized within the established scope as detailed by citing the function of MS-DOS and OS/2. At step 1210 performed when the event of the click is realized, the operation is peculiar to the present invention.

Then, referring to a flowchart shown in Figure 13, the content of step 1212 will be explained in more detail. At the outset, at step 1302, the current X coordinate and Y coordinate values for the reference point of the mouse cursor is acquired and then prepared as values, x and y, respectively.

At step 1304, the number of multiple pointing spots is prepared as value N. At step 1306, a variable i for the index is initialized to 1. At step 1308, SEL which is a variable showing at which point operation is selected is initialized to such values as -1 and 999 (hereinafter referred to as INVALID 1) which are not used as a number of the multiple pointing spots. At step 1310, a variable named an object ID stores such values as -1 and 999 which are not used as an actual object (hereinafter referred to as INVALID 2).

At step 1312, a determination is made as to whether or not i is less than N. When i is less than N, it is checked whether or not an object is present at a coordinate point of (x + dx_i, y + dy_i). At the initial step, a value i=1 is given, this is a determination as to which object the first pointing spot of the mouse cursor designates. Here, Symbols dx_i and dy_i designate the offset of the X coordinate and the Y coordinate from a reference point of the i-th point in the currently used multiple point mouse cursor obtained at step 1202 in Figure 12. It is determined whether or not an object is present in the coordinate by referring to the coordinate position of each object in the object control table shown in Figure 4.

At step 1314, when it is determined an object is present at coordinate (x + dx_i, y + dy_i), ID of the object is stored and i is stored in SEL followed by returning to the flowchart shown in Figure 12. The program shown in Figure 12 serves to bring back the object ID and SEL to some application program when the program shown in Figure 12.

After returning to Figure 13, when it is determined that no object is present at coordinate (x + dx_i, y + dy_i) at step 1314, i is increased by 1 at step 1320. Then the control returns to step 1312 to determine again

whether or not i is N or less. When i is N or less, the process proceeds to the above step 1314 onward.

On the other hand, when the determination at step 1312 is negative, this means that no object was found in any pointing spot $i=1$ through N . In such a case, the control returns to step the flowchart shown in Figure 12 with INVALID stored in the object ID.

Figure 14 is a view showing an example of a typical application program employing a mouse control routine shown in Figure 12. At step 1402, the application program uses the multiple point mouse changing apparatus 109 to select and set an appropriate multiple pointing spots mouse cursor from a mouse information control table 117.

At step 1404, when it is assumed that the system is represented in C, the application program calls a program and routine shown in the flowchart shown in Figure 12 with an address of a structure including, as elements thereof, the object ID and selection number SEL. The routine shown in Figure 12 then performs the transfer of the mouse cursor and detects the mouse click event. At step 1406, it is determined whether or not an object ID value sent back from the routine shown in Figure 12 is INVALID 2. As is apparent from Figures 12 and 13, the fact that an object ID value is INVALID 2 means that either no mouse click event is generated (refer to step 1210) or no object is present at any position of the multiple pointing spots of the mouse cursor (refer to steps 1310 and 1314). In such a case, the application program does not do anything.

When the object ID value is sent back from the routine shown in Figure 12 are not INVALID 1, that is to say, when the mouse is being clicked, it is determined at step 1406 which of the multiple pointing spots of the mouse cursor designates which of the objects. When such determination is made, the control proceeds to the determination of the SEL value sent back from the routine shown in Figure 12 (step 1408). As mentioned above, the SEL value is the number of the pointing spot designating the object. When the SEL value is an appropriate number, operation is performed in accordance with the value at step 1410. Incidentally, steps 1408 and 1410 are described as follows when a C like pseudo code is used.

```

switch (SEL)    {
case 1:  operation 1;
          break;
5  case 2:  operation 2;
          break;
10 case 3:  operation 3;
          break;
          . . . . .
          . . . . .
15 case N:  operation N;
          break;
default:  exceptional operation;
          break;
20 }

```

Operations 1 through N as well as exceptional operations are to be preliminarily designated by the person who prepares the application. Incidentally the application program does not call the routine shown in Figure 12. As described later with reference to Figure 15, the mouse information control table 115 may be directly accessed through the multiple pointing spots changing apparatus 109 to acquire and process offset information for multiple pointing spots.

Application Example

The multiple point mouse cursor of the present invention is widely useful for various interactive operations. Although some application examples of the multiple point mouse cursor of the present invention will be shown hereinbelow, it is to be understood that the scope of application is not limited to them.

1. Designation of Vertical Bisector of Line

A certain kind of education program effects an operation for positioning one of two lines as a vertical bisector relative to the other. However, with conventional mouse cursors, it is difficult to designate which should be a bisector line and which should be a line to be divided into two by the bisector. For example, it may be possible to designate a line designated first by clicking to be the line to be divided into two and a line designated later to be the bisector. However, some positional relationship between the two lines disturbs orderly relations, thereby confusing users to a considerable degree in operation.

The operation shown in Figure 15 solves this problem. At the outset, it is to be assumed that two point cursor shown in Figure 9 is designated. At step 1502, variables such as Segment_1 and Segment_2

store a value such as INVALID. The meaning of INVALID is the same as that mentioned with respect to step 1310 shown in Figure 13. At step 1504, the mouse click is checked so that when the mouse click is detected, the coordinate positions of p1 and p2 shown in Figure 9 are stored as (xp1, yp1) and (xp2, yp2). In more detail, the operation is effected in the following way. In other words, when (xp1, yp1) is set as the reference point, these values can be immediately acquired with the aforementioned software interruption or API calling. In addition, the mouse information control table stores offset X coordinate and Y coordinate values from P1 to P2. Consequently, addition of the offset values to the reference value allows immediate calculation of P2 coordinate value.

At step 1508, it is checked whether or not Segment_1 is INVALID. If Segment_1 is INVALID, it means that no line portion is registered. Thus, at step 1510, it is checked whether or not the coordinate (xp1, yp1) is located on the line portion (the line portion is considered as an object here only). If the coordinate is located on the line portion, the line portion is registered in Segment_1 at step 1512.

In the same way, step 1514 is to check whether or not Segment_2 is INVALID. If Segment_2 is INVALID, it means that Segment_2 registers no line portion. Thus step 1516 is to check whether (xp2, yp2) is located on a line portion (the line portion is considered as an object here only). When it is found that (xp2, yp2) is located on the line portion, Segment_2 registers the line portion at step 1518.

Step 1520 is to check whether or not either Segment_1 or Segment_2 is still INVALID. If either Segment_1 or Segment_2 is INVALID, the control returns to the determination at step 1504.

The fact that neither Segment_1 nor Segment_2 is INVALID means that both Segment_1 and Segment_2 register an appropriate object ID for an appropriate line portion. Thus at step 1522, combinations of line portions are treated by regarding Segment_1 as a line vertically divided into two with a bisector and Segment_2 as a bisector vertically dividing into two the former line portion.

Next, referring to Figure 16, the action of the flowchart shown in Figure 15 will be detailed. Figure 16 (a) shows line portions L1 and L2 and double point cursor 1602 on the screen.

In Figure 16 (b), line portion L1 is clicked at pointing spot P1 of the double point cursor 1602, thereby registering line portion L1 as Segment_1.

In Figure 16 (c), line portion L2 is clicked at point P2 of the double point cursor 1602, thereby registering line portion L2 as Segment_2.

In Figure 16 (c), the registration of line portion L1 and line portion L2 sets L1 as a line vertically divided with the bisector and L2 as a bisector vertically dividing the former line, thereby combining the two line portions. At this time, L1 is clicked with P1, whereas

L2 is with P2. In such a case the result of the operation is unrelated to the order of the clicking. Users can thus reduce any annoyance of the operator.

2. Copy and Transfer of Object

Referring to the flowchart in Figure 17, copy and transfer operation of objects will be detailed. In such a case, assume that a double point cursor is set as shown in either of Figure 7 or in Figure 9. At the outset, step 1702 is to set COPY flag to 1. At step 1704, a value INVALID is set to a variable Object ID. This operation is similar to step 1502 shown in Figure 15. Step 1706 is to check the mouse clicking. When the presence of mouse clicking is detected, p1 and p2 coordinate positions shown in Figure 9 are stored as (xp1, yp1) and (xp2, yp2), respectively at step 1708.

Step 1710 is to check whether or not the formula Object ID=INVALID is established. If the formula is established, a copy of object or the object to be moved are not yet selected. The process thus proceeds to step 1712 to determine whether or not an object is present at (xp1, yp1). If an object is present at such a coordinate position, Object ID registers the ID of the object at step 1714 to return to step 1706. If an object is absent, a determination is made at step 1716 as to whether an object is present at (xp2, yp2). If an object is present at (xp2, yp2), Object ID registers the ID of the object at step 1718. At the same time, the COPY flag is set to 0 at step 1720 and returns to step 1706.

On the other hand, if the determination of Object ID=INVALID is negative at step 1710, the process proceeds to step 1722 to check the COPY flag value. When COPY flag=1 is established, copy operation is performed at step 1724. If it is not established, transfer operation is performed at step 1726.

The state of operation is explained in conjunction with Figure 18. Figure 18 shows two point cursor 1802 having pointing spot P1 and pointing spot P2, and an object (triangular figure 1804). In Figure 18 (a), when an object 1804 is clicked at pointing spot P2, the object is registered as an object to be processed. At the same time, the clicking of the object at pointing spot P2 causes COPY flag to assume 0, thereby selecting a transfer operation. Then, as shown in Figure 18 (b), when the mouse is clicked by moving a cursor, the object 1804 is moved to the current position of the cursor.

Then, in Figure 18 (c), when the object 1804 is clicked at pointing spot P1, the object is registered as the object to be processed. At the same time, the clicking of the mouse cursor at pointing spot P1 causes the COPY flag to assume 1 to select copying operation. Subsequently, as shown in Figure 18 (c), the clicking of the mouse by moving the cursor allows copying operation of object 1804 at the current position of the cursor, thereby generating a new object 1806. Incidentally, though not shown in the drawings,

character C may be represented at pointing spot P of the cursor 1802 and character M at pointing spot P2 so as to clarify the copying and transferring operations.

3. Selection of Angle

As shown in Figure 19, when line portions AB and BC determine angles α and β , merely clicking line portions AB and BC in sequence cannot clearly determine which angle should be selected with one point mouse cursor in an attempt to draw a bisector of an angle relative to the angles. Thus it is possible to conceive of a technique of designating an angle clockwise from the line portion initially clicked followed by designating an angle to the line portion clicked afterwards. However, when the line portion is complicated, it is very difficult for an operator to designate the order of clicking line portions without error. Then, in accordance with the present invention, the double point cursor 1902 is used. In such a case, each pointing spot P1 and P2 preferably represent an arrow indicating the designation direction as shown in Figure 20.

In Figure 19 (b), line portion AB is clicked with pointing spot P1. In Figure 19 (c), the click of line portion BC allows selecting an angle α to give a bisector of the angle in Figure 19 (d). Incidentally, even when the line portion BC is initially clicked with the pointing spot P2 followed by clicking the line portion AB with the pointing spot P1, the result is the same. The flow-chart of this operation may be the same as the counterpart shown in Figure 15. Thus explanation thereof is omitted here.

4. Other Application Examples

In addition to the above applications, the following applications can be considered.

(1) The double point cursor is used to select an object with one pointing spot to select and execute the object with another pointing spot (in place of double clicking).

(2) The double point cursor is used to select a single object with one pointing spot and to then select a plurality of objects with the other pointing spot. For a plurality of selected objects, batch deletion, batch printing, and batch editing are possible.

(3) The triple point cursor is used to minimize the window with one pointing spot whereas the other triple point cursor is used to maximize the window. The remaining pointing spot is used to restore the window.

(4) In the text editing application, the double point cursor is used to designate the initiation of input for insertion from the cursor position with one pointing spot and the initiation of input for replacement with another pointing spot.

(5) In the text editing application, each pointing spot in the n-th point cursor is used for setting the Roman alphabet input mode, Kana input mode, full-size alphanumeric input mode, and half-size alphanumeric input mode, and half-size Kana input mode.

(6) In the text editing application, for such operation as transfer, copy and change of attributes line unit is selected with one pointing spot and character unit is selected with the other pointing spot.

(7) In the text editing application, each pointing spot in the n-th point cursor is used for the selection of different fonts such as Gothic, Courier, and Elite.

(8) In the figure and graphic editing application, the double point cursor is used to move an object in parallel with one pointing spot and to rotate and move an object with the other pointing spot.

(9) In the figure and graphic editing application, the double point cursor is used to rotate an object through a voluntary angle with one pointing spot and to invert an object by a specific angle.

In the figure and graphic application, when two over-lapping objects are present, a two point cursor is used to arrange one pointing spot before the lower object and the other pointing spot behind the upper object.

(10) In the figure and graphic editing application, the n-th point cursor is used to change the attributes of figures such as thickness of lines, kinds of line, paint patterns of surfaces, and colors.

(11) In the figure and graphic editing application, a triple point cursor is used to select pointing spots, lines and surfaces on each line.

(12) In the figure and graphic editing application, a double point cursor is used to designate two parallel line portions at each pointing spot. (The length of the line portion clicked at the second pointing spot is changed so as to agree with the length of the length of the line portion clicked at the first pointing spot.)

(13) In the figure and graphic editing application, a double point cursor is used to designate two line portions whose lengths are to be matched at each pointing spot. (The line portion clicked at the second pointing spot is changed so as to agree with the length of the line portion clicked at the first pointing spot.)

(14) In the figure and graphic editing application, a double point cursor is used to designate two angles whose degrees are to be matched at each pointing spot. (The degree of the angle clicked at the second pointing spot is changed so as to agree with the angle clicked at the first pointing spot.)

(15) In the figure and graphic editing application, a double point cursor is used to designate two

pointing spotting spots of objects to be overlapped at each pointing spot. (The object clicked with the second pointing spot is moved so as to agree with the object clicked at the first pointing spot.)

(16) In the figure and graphic editing application, a two point cursor is used to designate a line portion with one pointing spot and a surface with the other pointing spot to designate an angle between the line portion and the surface (in a three-dimensional case).

(17) In the figure and graphic editing application, a two point cursor is used to designate a surface with one pointing spot and another surface with the other pointing spot to designate an angle between the two surfaces (in a three-dimensional case).

That is to say, a mouse is outlined as a pointing device in the above explanation. However, the present invention is not limited only to a mouse as a pointing spotting device. It is to be understood that the invention can be applied to a system having a voluntary pointing device such as a track ball or joy stick with a cursor that cooperatively moves on a screen.

In addition, although the above explanation describes that the above embodiment uses established API such as MS-DOS and OS/2, it is possible for a creator of application programs to follow the teachings of the invention to prepare a unique routine for controlling a mouse cursor. In such a case, it is possible to prepare a mouse cursor having a sufficient size to apply the cursor to applications, which is more advantageous for the multiple point cursor.

As described above, the present invention uses a multiple point cursor to click an object at different pointing spots as needed, thereby enabling an operator to execute a desired operation with a minimum of operation without requiring complicated mouse operations such as double clicking and dragging.

Claims

1. A data processing system having a processor, a display coupled to the processor and a pointing device coupled to the processor such that movement of the pointing device results in corresponding movement of a cursor on the screen of the display, the data processing system comprising:

means for displaying a cursor having first and second pointing spots on the screen of the display;

means for instructing the processor to perform a first operation represented by an object displayed on the screen when the first pointing spot of the cursor is positioned on that object; and

means for instructing the processor to perform a second operation represented by an object displayed on the screen when the second pointing

ing spot of the cursor is positioned on that object.

2. A data processing system as claimed in claim 1 wherein the pointing device is a mouse having at least one button.

3. A data processing system as claimed in claim 2 wherein the means for instructing the processor include a click operation on a button of the mouse.

4. A data processing system as claimed in any of the preceding claims wherein the second operation is different from the first operation.

5. A data processing system as claimed in any of the preceding claims further comprising:

means for instructing the processor to register an object as a first object to be processed when the first pointing spot is positioned on that object on the screen; and

means for instructing the processor to register a different object as a second object to be processed when the second pointing spot is positioned on a different object from the first object; and

means responsive to the registration of the first object and the second object for instructing the processor to perform a predetermined operation which relates the first object to the second object.

6. A data processing system as claimed in any of the preceding claims comprising:

means for associating each operation to be performed by the processor with each of the pointing spots; and

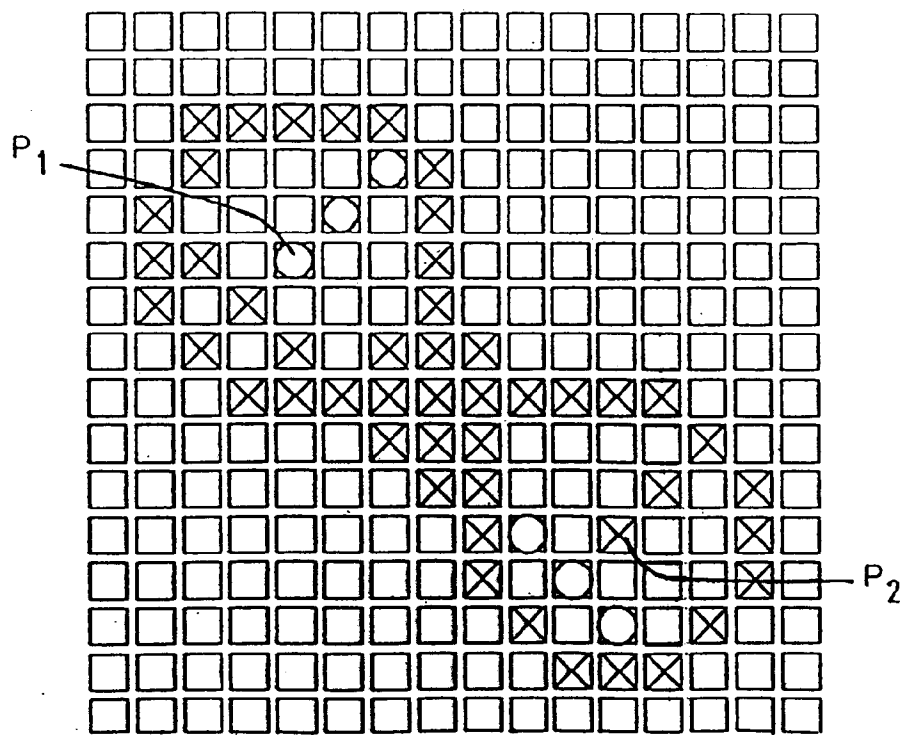
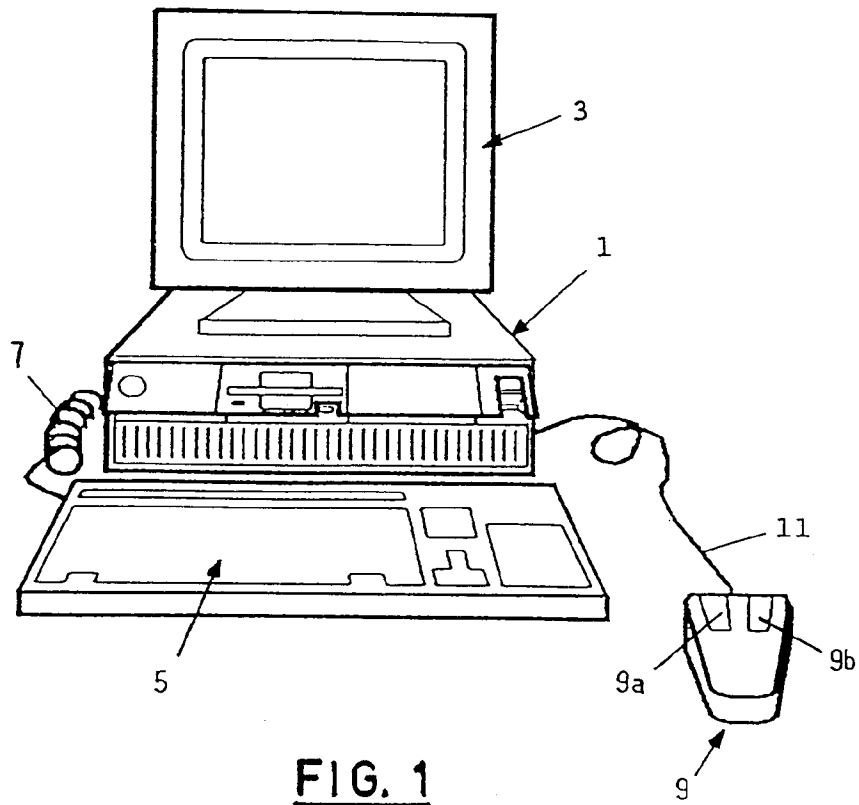
means for determining if any of the pointing spots is positioned on an object displayed on the screen.

7. A method of operating a data processing system having a processor, a display coupled to the processor and a pointing device coupled to the processor such that movement of the pointing device results in corresponding movement of a cursor on the screen of the display, the method comprising the steps of:

displaying a cursor having first and second pointing spots on the screen of the display;

instructing the processor to perform a first operation represented by an object displayed on the screen when the first pointing spot of the cursor is positioned on that object; and

instructing the processor to perform a second operation represented by an object displayed on the screen when the second pointing spot of the cursor is positioned on that object.



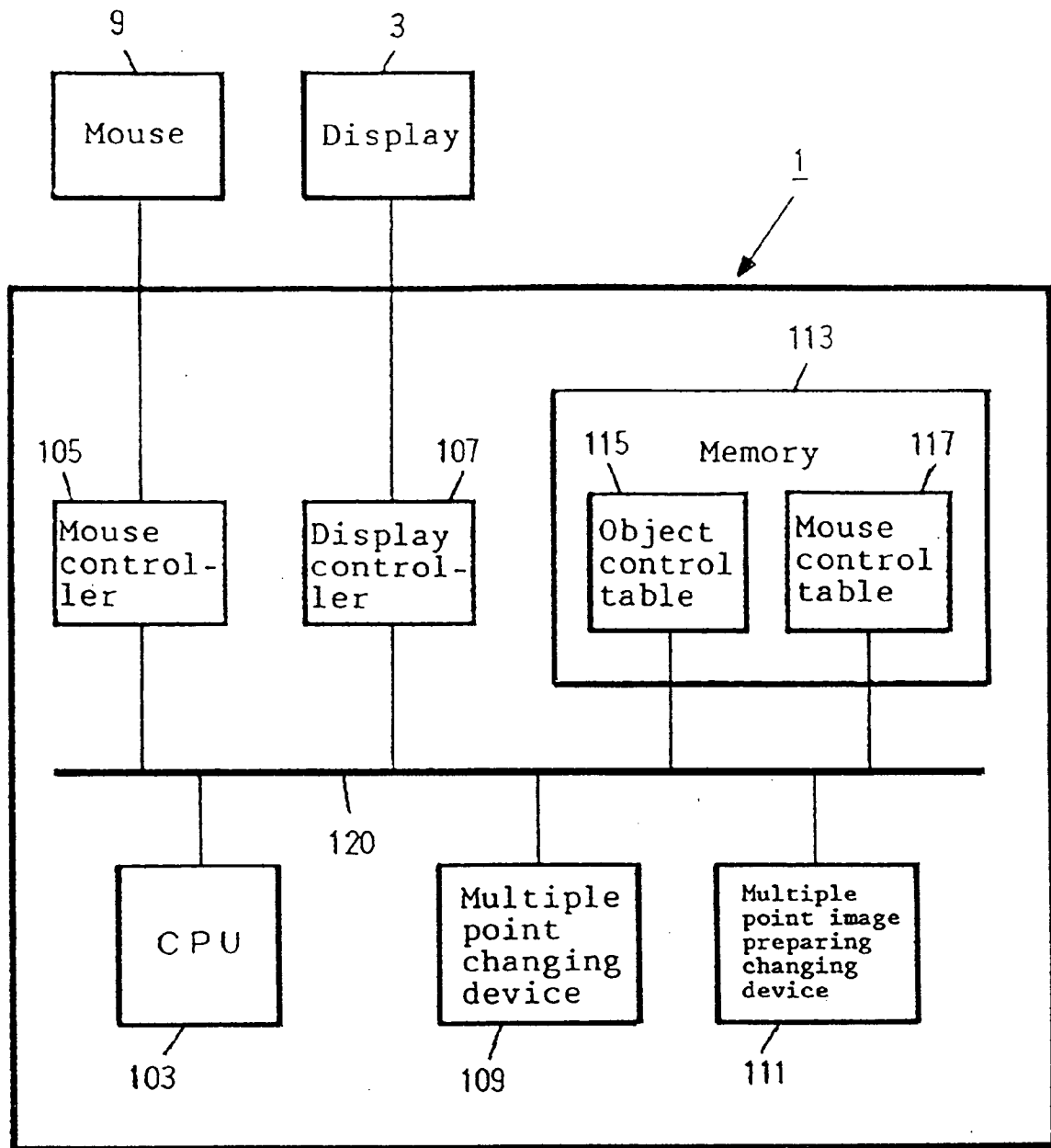


FIG. 2

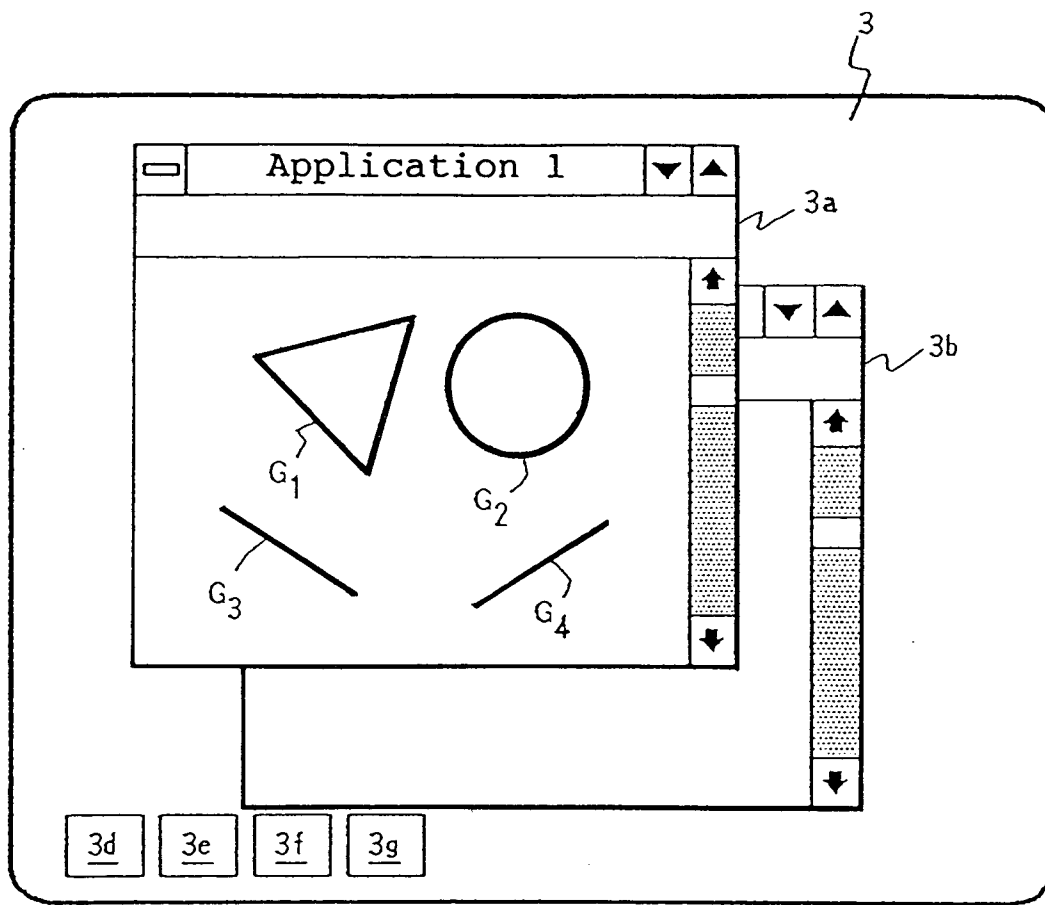


FIG. 3

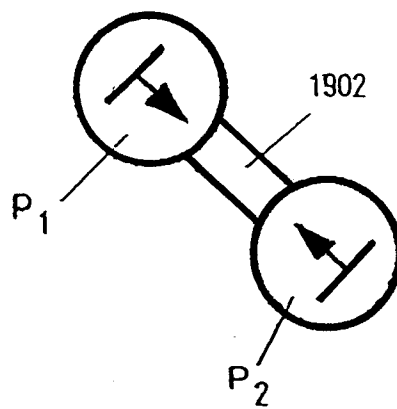


FIG. 20

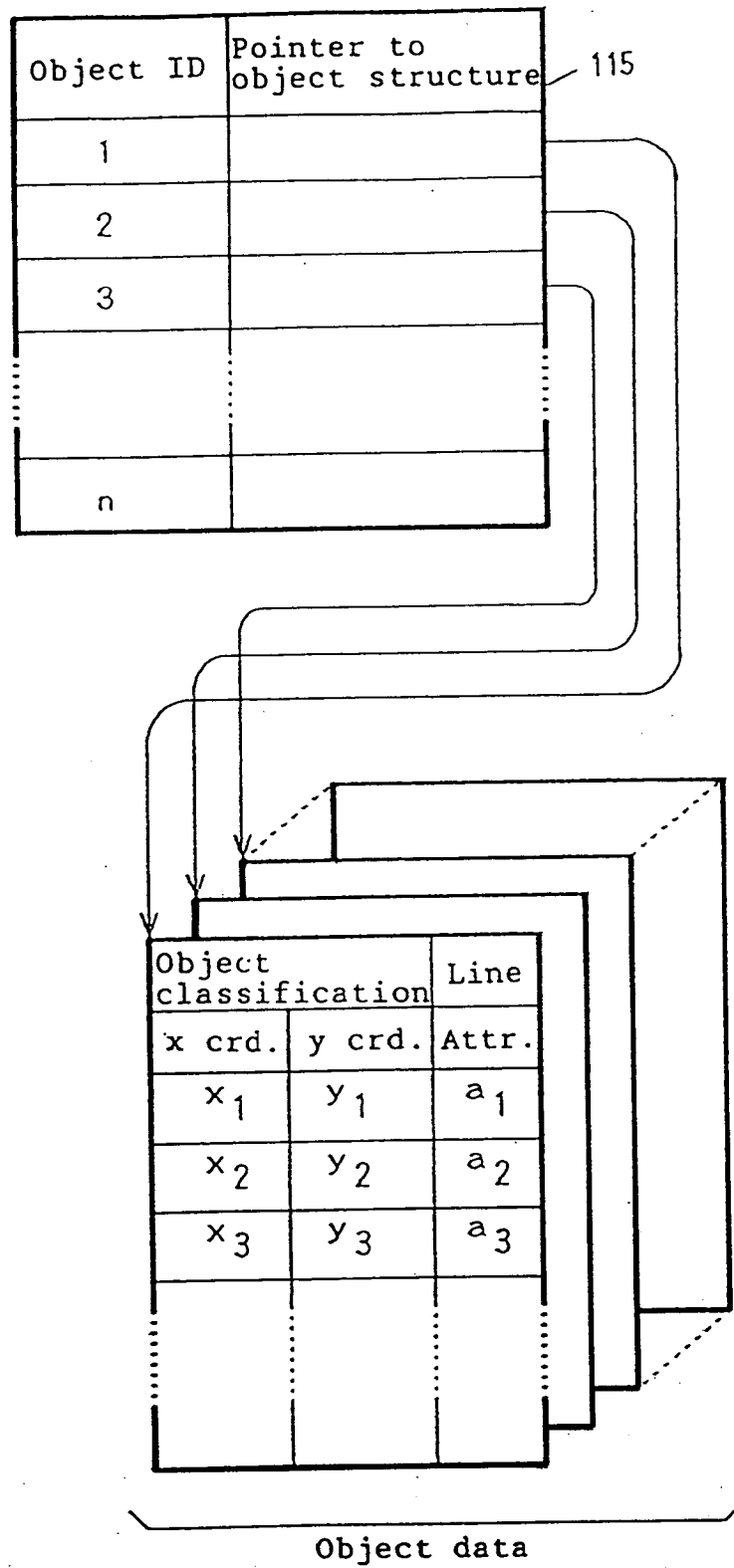


FIG. 4

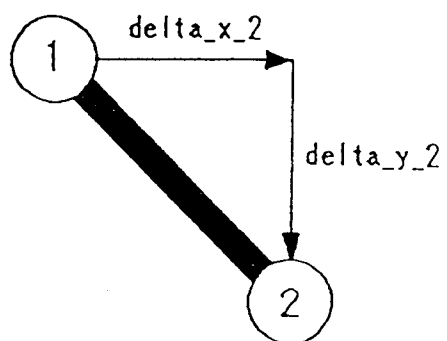
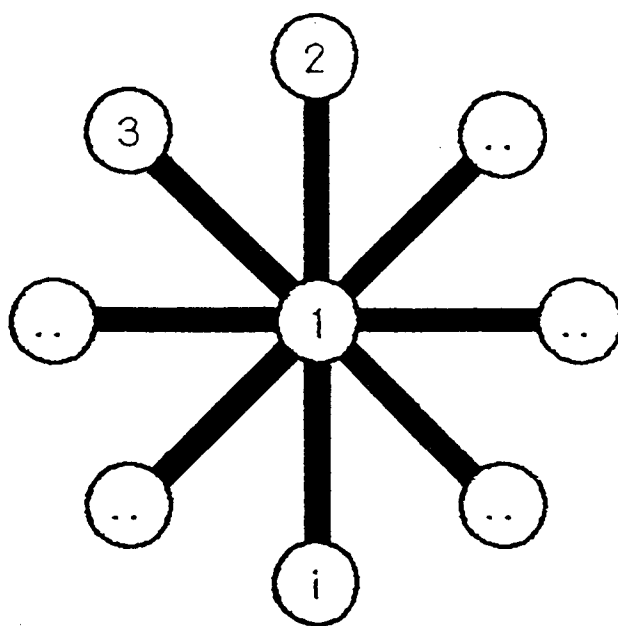


FIG. 5



$$X_i = X + \text{delta_x_i}$$

$$Y_i = Y + \text{delta_y_i}$$

FIG. 6

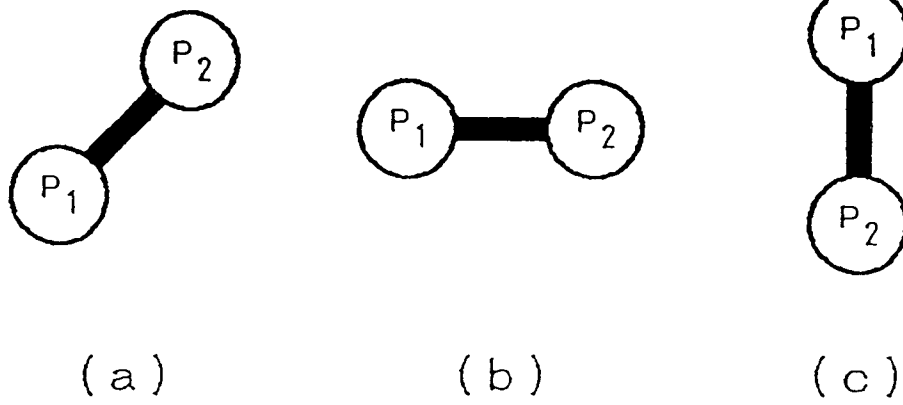


FIG. 7

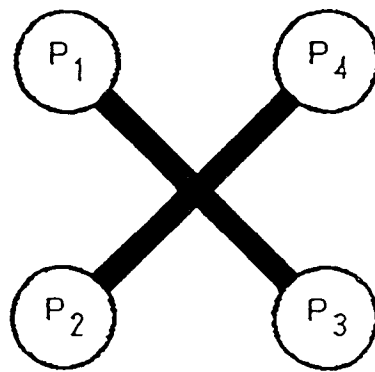


FIG. 8

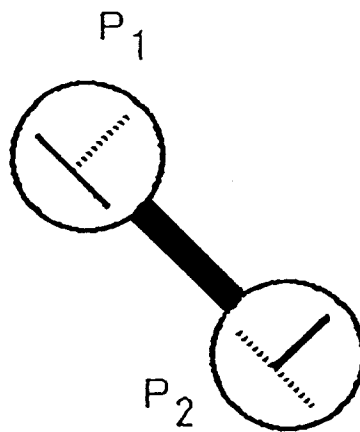
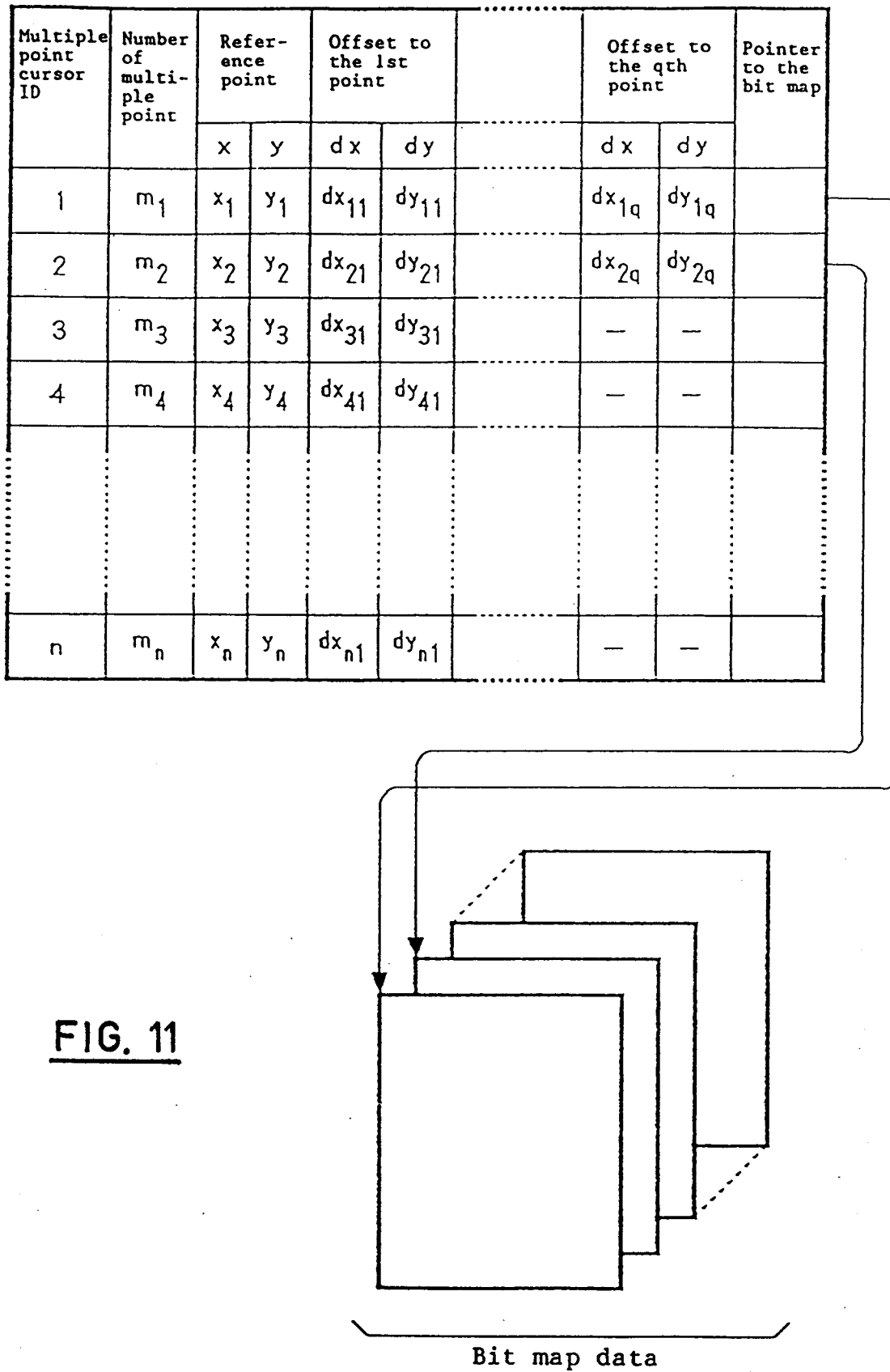
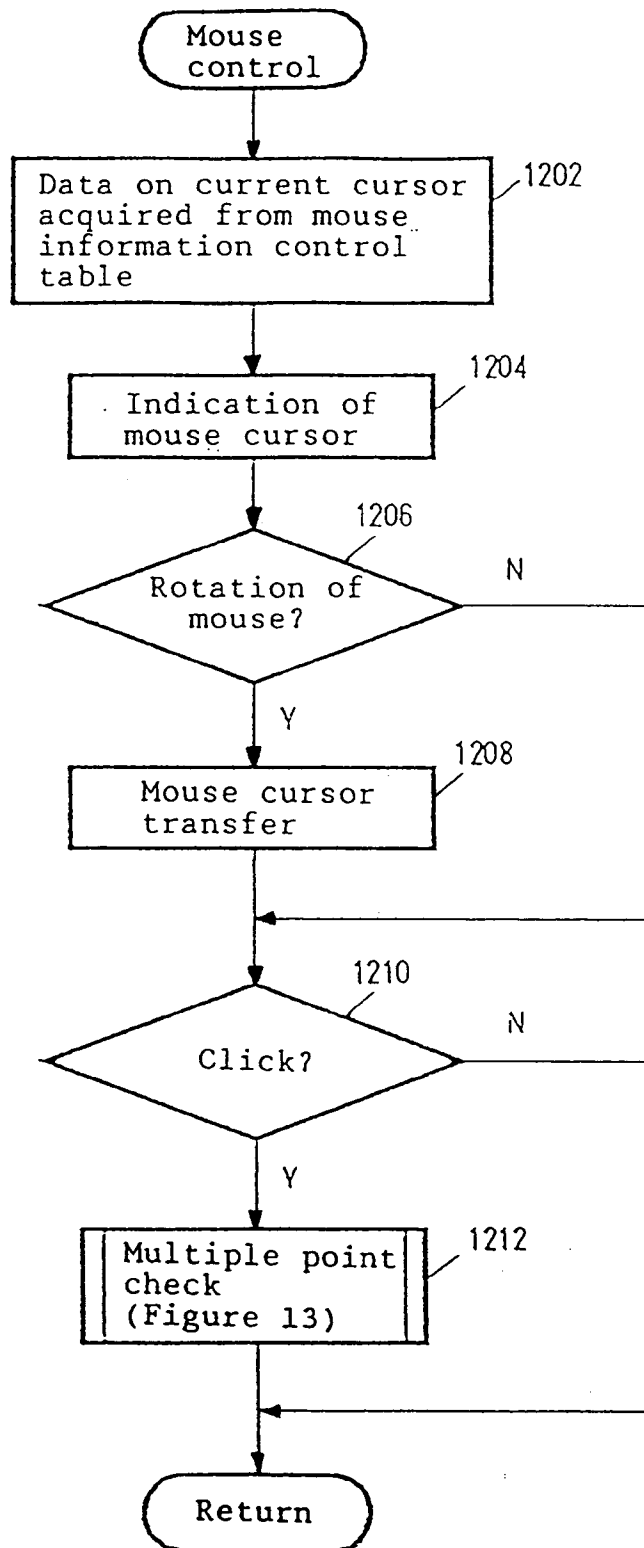
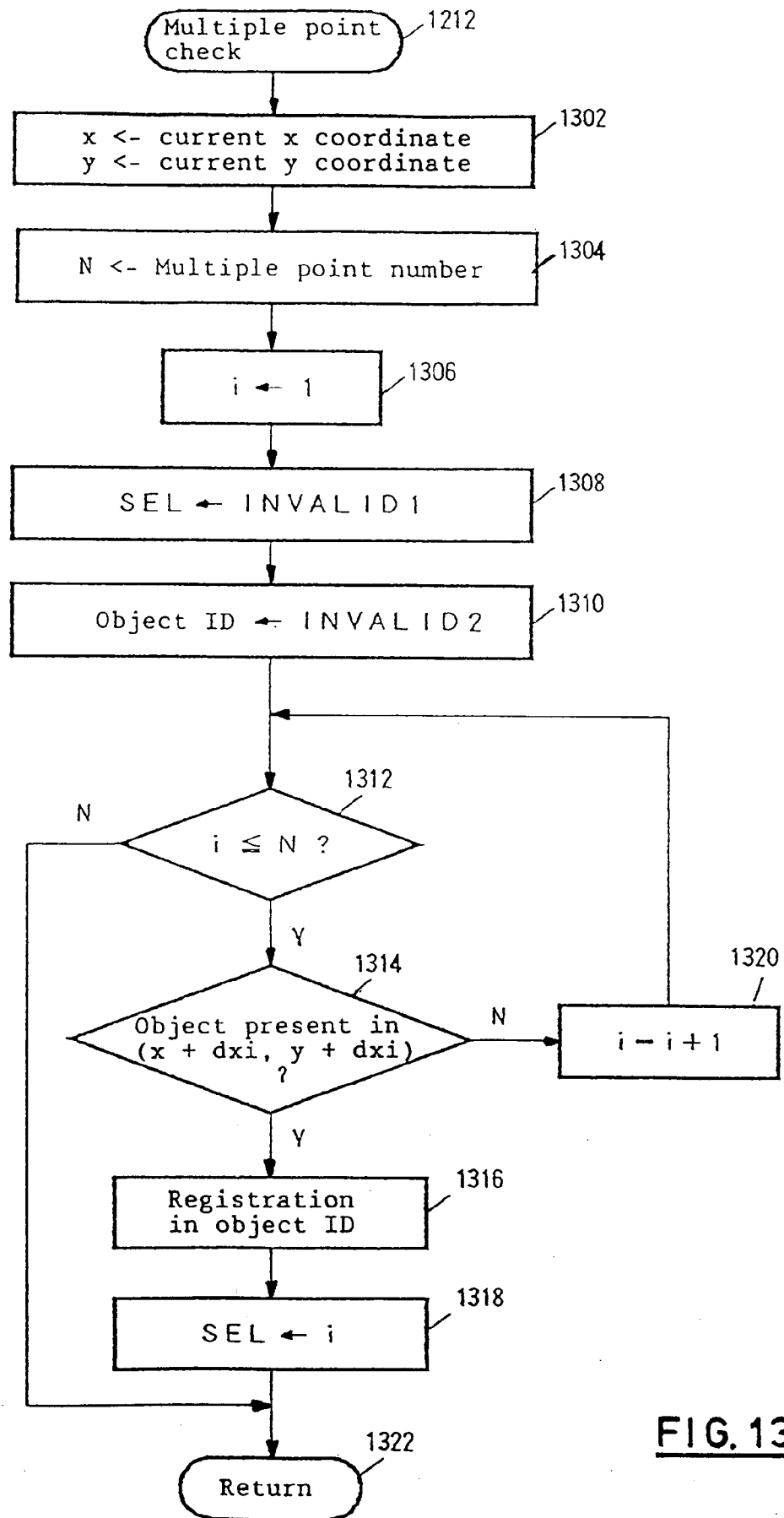
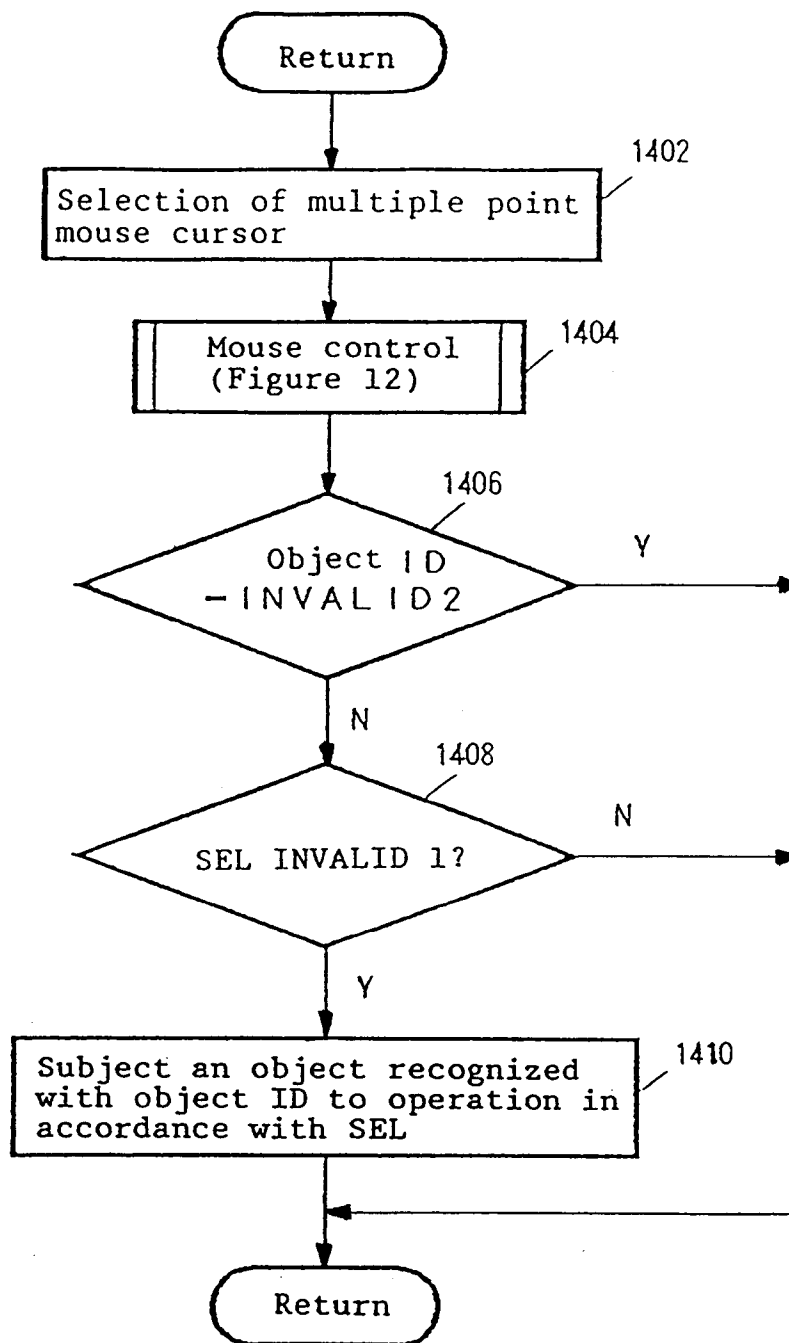


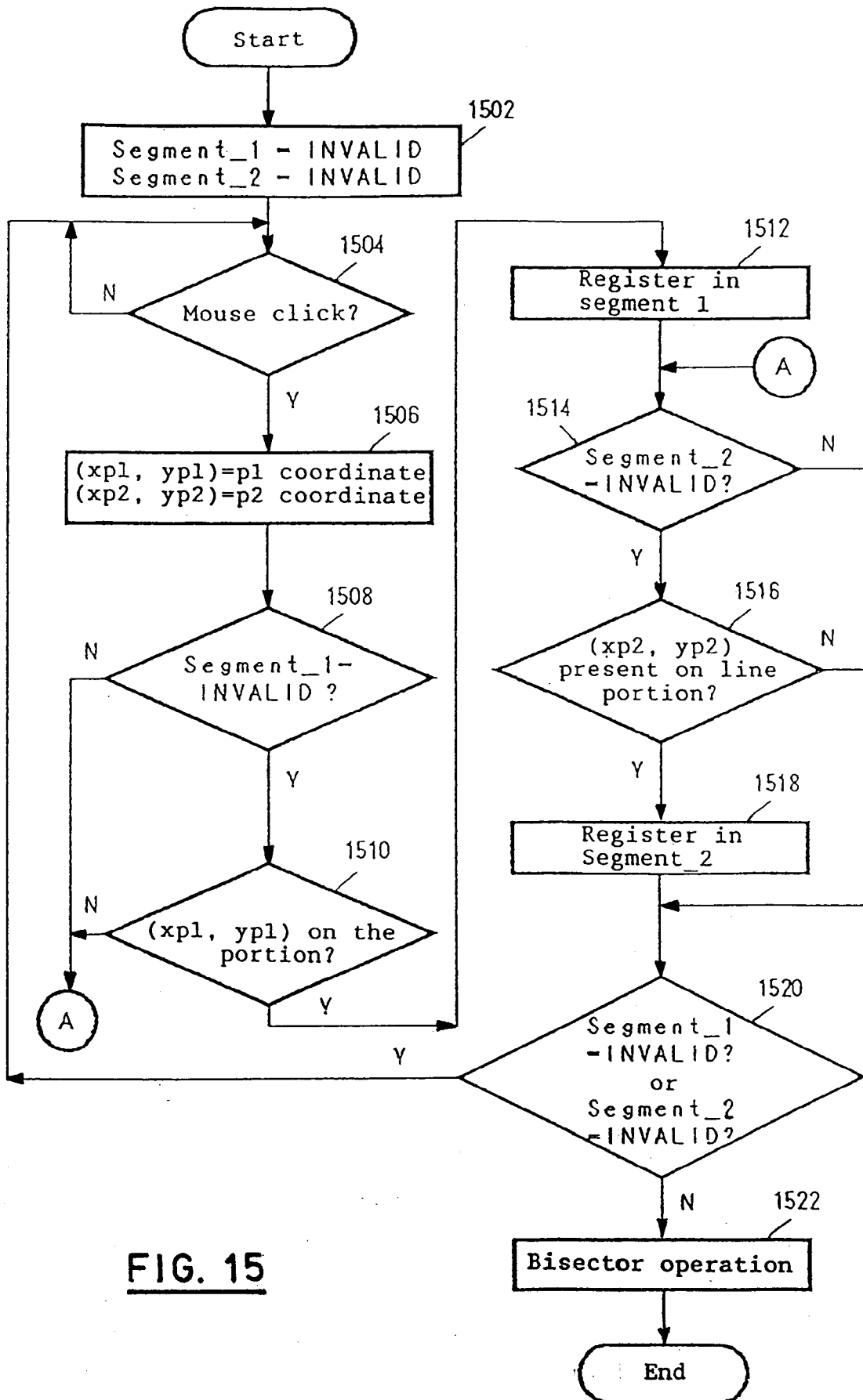
FIG. 9

**FIG. 11**

**FIG. 12**

**FIG. 13**

**FIG. 14**

**FIG. 15**

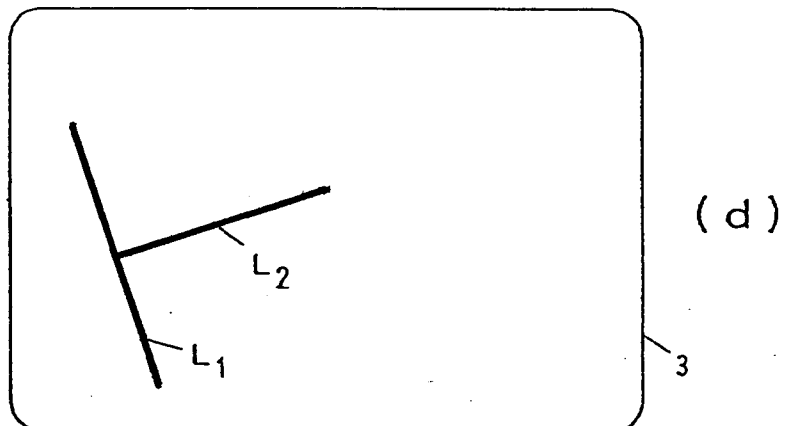
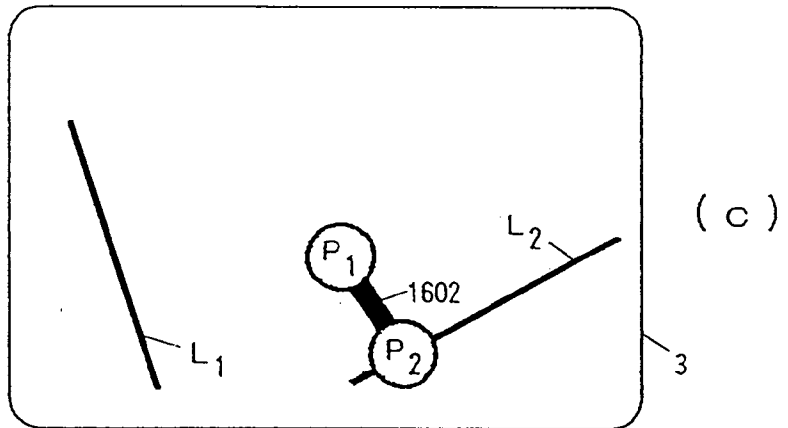
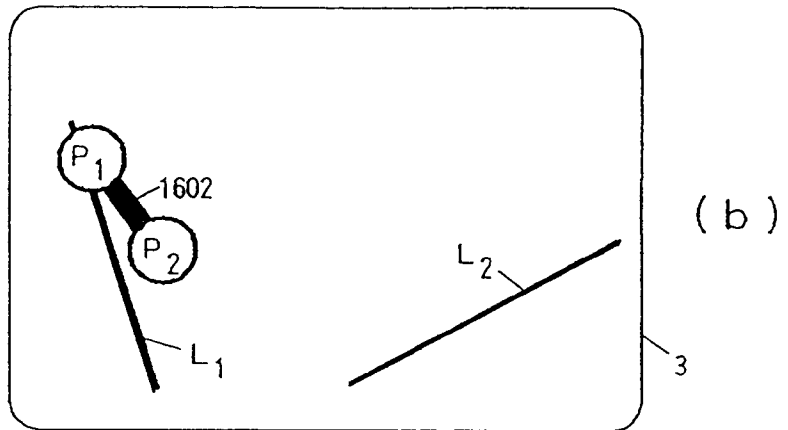
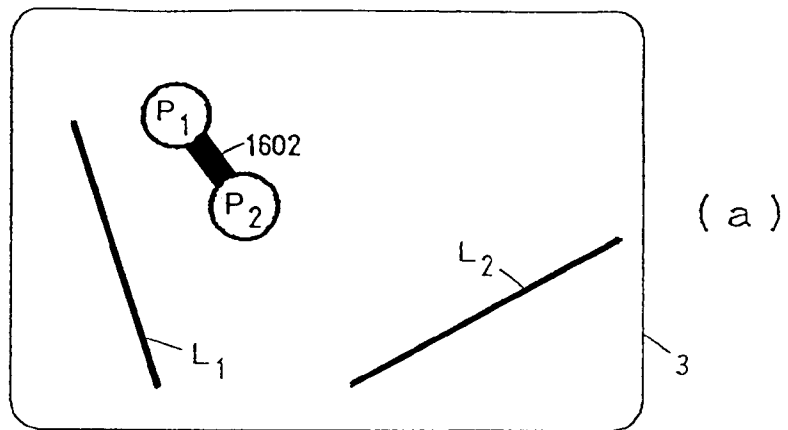


FIG. 16

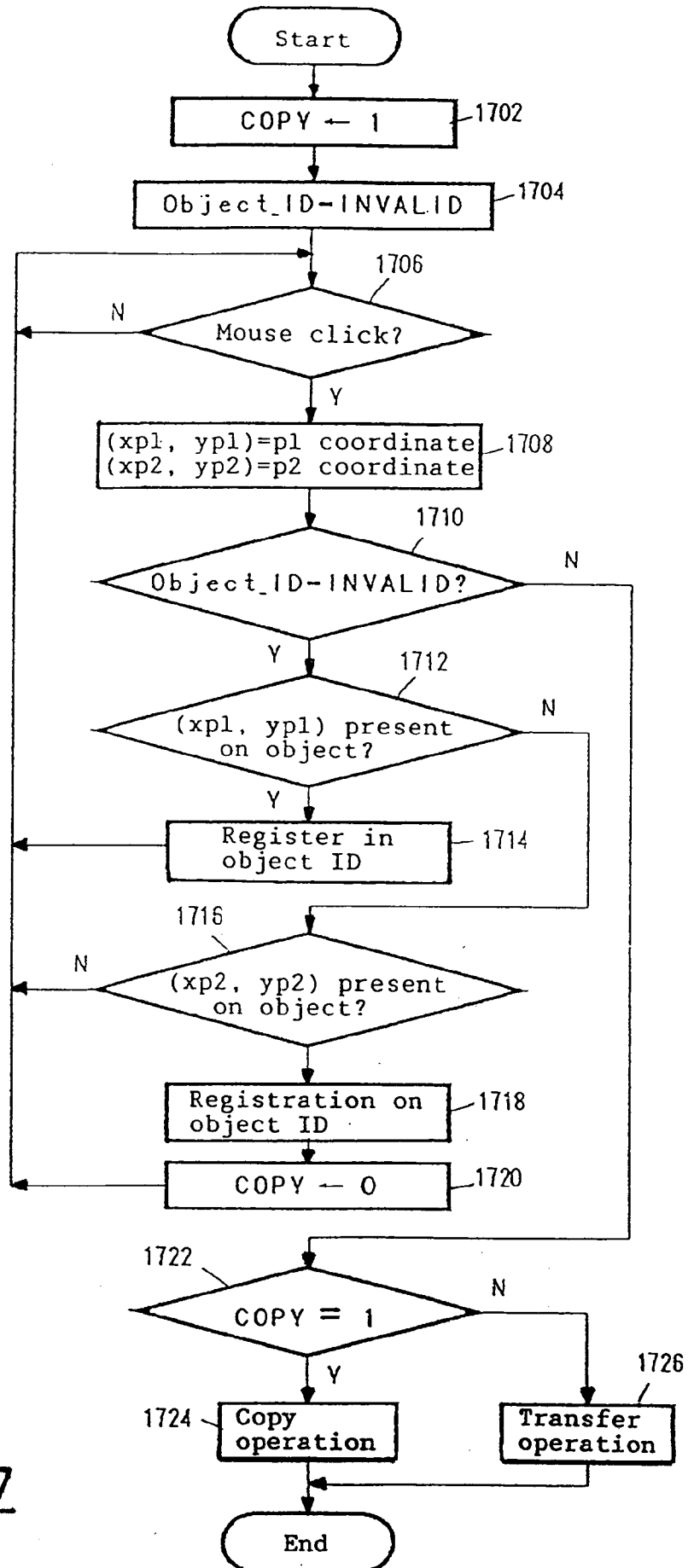
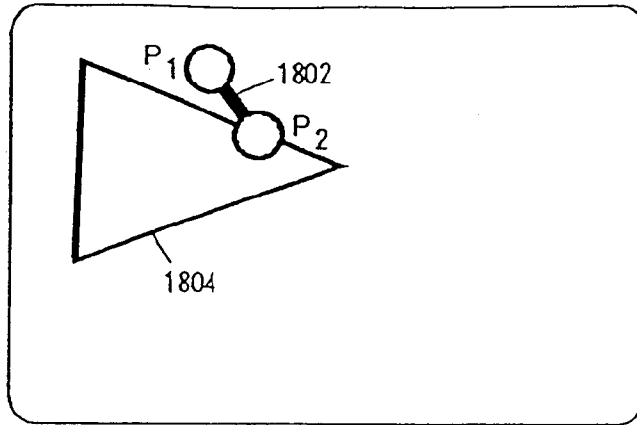
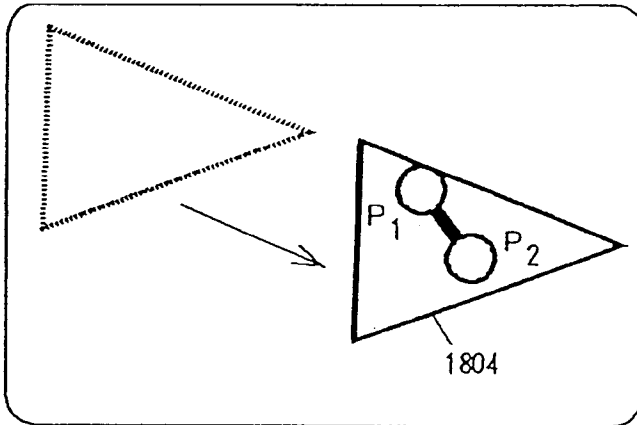


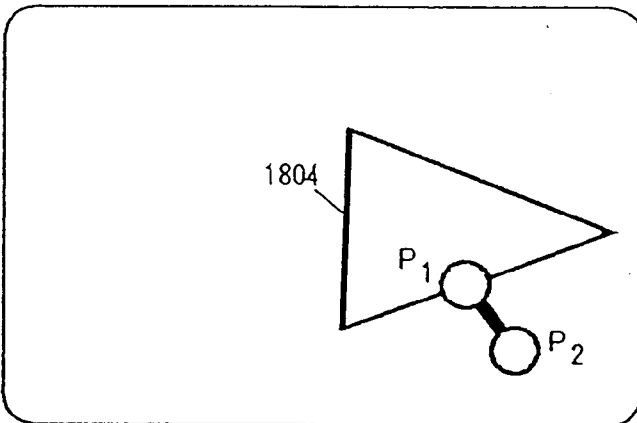
FIG. 17



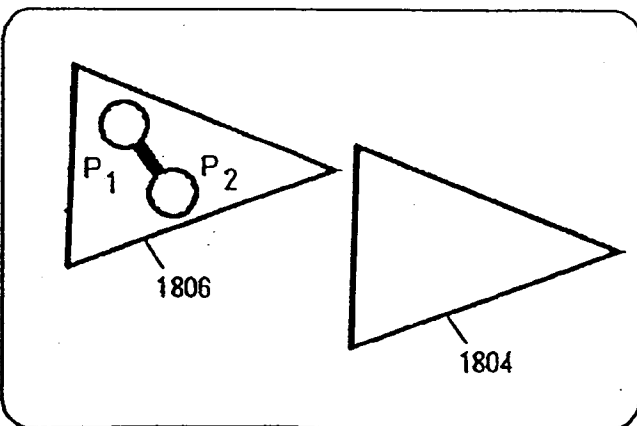
(a)



(b)



(c)



(d)

FIG. 18

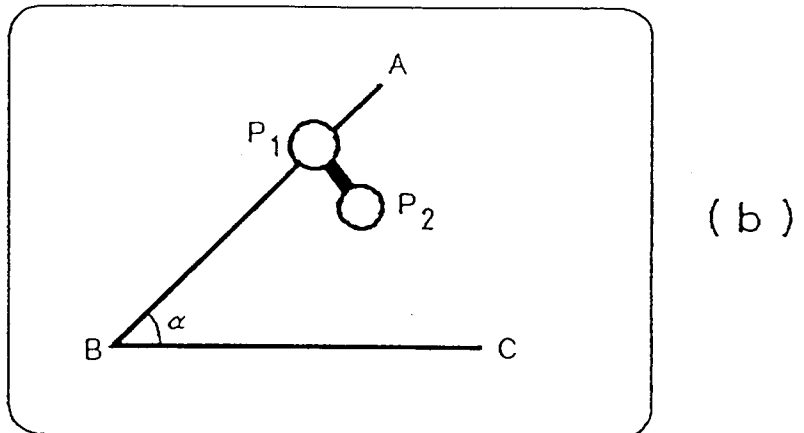
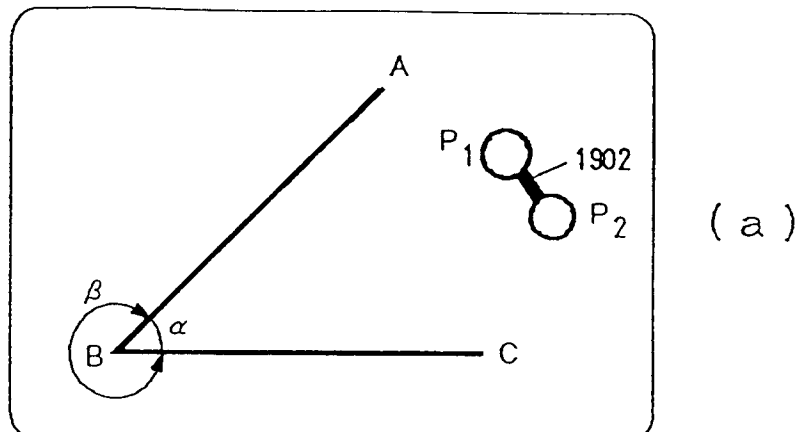


FIG. 19

